UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/815,904	03/31/2004	Eric F. Vannerson	42P19126	9294

8791 7590 08/22/2008

BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP
1279 OAKMEAD PARKWAY
SUNNYVALE, CA 94085-4040

EXAMINER	
HUISMAN, DAVID J	

ART UNIT	PAPER NUMBER
2183	

MAIL DATE	DELIVERY MODE
08/22/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/815,904	VANNERSON ET AL.	
	Examiner	Art Unit	
	DAVID J. HUISMAN	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *02 June 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,2,5-8,10-19,21-26,28 and 29* is/are pending in the application.

4a) Of the above claim(s) *10-17* is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2,5-8,10-19,21-26,28 and 29* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *02 June 2008* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-2, 5-8, 18-19, 21-26, and 28-29 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Amendment as received on 6/2/2008.

Specification

3. As admitted by applicant, and now according to at least amended claim 1, the three debug register bit fields are part of the processor control status register. As previously discussed, these fields are not added or attached to the control status register. Hence, applicant is requested to locate all portions of the specification (not the claims) which discuss adding/attaching these fields and change the language appropriately.

Election/Restrictions

4. This application contains claims drawn to an invention nonelected with traverse in the reply filed on June 2, 2008. A complete reply to the final rejection must include cancellation of nonelected claims or other appropriate action (37 CFR 1.144) See MPEP § 821.01.

Claim Objections

5. Claim 1 is objected to because of the following informalities:

- Please either replace "at least one" with --a-- in line 6, or insert --at least one-- before “breakpoint” in lines 7 and 9.
- In line 9, please replace “that the" with --to which--.

Appropriate correction is required.

6. Claim 18 is objected to because of the following informalities:

- Please either replace "at least one" with --a-- in line 4, or insert --at least one-- before “breakpoint” in lines 5 and 7.
- In line 7, please replace “that the" with --to which--.
- In the 4th to last line, either insert a comma before “each” or replace “each comprising” with --that each comprise--

Appropriate correction is required.

7. Claim 19 is objected to because of the following informalities: Please either replace “said breakpoint bit” with --said at least one breakpoint bit field-- in line 3, or insert --field-- after “bit” in line 3 and delete “at least one” in the last two lines (depending on the correction chosen for claim 18). Appropriate correction is required.

8. Claim 24 is objected to because of the following informalities:

- Please either replace "at least one" with --a-- in line 2, or insert --at least one-- before “breakpoint” in lines 3 and 5.
- In line 5, please replace “that the" with --to which--.
- In the 6th to last line, either insert a comma before “each” or replace “each comprising” with --that each comprise--

Appropriate correction is required.

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. Claims 1-2, 5-8, 18-19, 21-26, and 28-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Glew et al., U.S. Patent No. 5,694,589 (herein referred to as Glew), in view of IBM Technical Disclosure Bulletin NN8907370 (as previously cited and herein referred to as IBM), and further in view of Deng et al., U.S. Patent No. 6,951,416 (herein referred to as Deng).

11. Referring to claim 1, Glew has taught an apparatus comprising:

a) a memory. See Fig.1, component 17.

b) a plurality of processors coupled to the memory. See Fig.1, Fig.3, and Fig.4. Note that a superscalar processor includes N processing elements to execute up to N instructions in parallel (this is the inherent nature of a superscalar system). Hence, the N processing elements are the plurality of processors, as they each process data.

c) Glew has not taught a controller coupled to the memory and the plurality of processors, the controller to execute a debug process that attaches at least one breakpoint bit field comprising a single bit directly to one or more instructions of the plurality of processors, the breakpoint bit field to enable a user of the apparatus to set a breakpoint based on an address of an instruction of the one or more instructions that the breakpoint bit field is attached without having to perform an address comparison. However, IBM has taught such a concept. See Fig.2 and the disclosure.

Essentially, IBM allows a user to attach a breakpoint bit to each instruction without performing address comparison, thereby saving hardware, delay, and power associated with the comparison circuitry. One of ordinary skill in the art would have recognized that the controller of Glew (Fig.3, component 200, and column 6, lines 24-38) would be replaceable by the controller of IBM without "breaking" Glew because both Glew and IBM teach generating single breakpoint bits and attaching each bit to an instruction. In Glew, when the instructions are to be decoded and executed, all that is looked for is an instruction having a breakpoint bit. It doesn't matter whether that extra bit was attached in the manner specified by Glew or in the manner specified by IBM. Therefore, in order to maintain a system capable of debugging a system with multiple processors while reducing hardware, delay, and power consumption associated with Glew's address comparison circuitry, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Glew such that the controller executes a debug process that attaches at least one breakpoint bit field directly to one or more instructions of the plurality of processors, the breakpoint bit field to enable a user of the apparatus to set a breakpoint based on an address of the particular instruction without having to perform an address comparison.

d) Glew has further not taught that the debug process manipulates at least three debug register bit fields of at least one processor control status register, the at least three register bit fields comprising a run field, a single step field and a debug enable field that each comprise a single bit. However, Deng has taught such a concept. See Fig.16 of Deng and note field 208, which is a debug enable field, field 206, which is a single-step field, and fields BE1, BE2, BE3, or BE4, any one of which is a run field. These bits allow for increased functionality and flexibility in debugging. For instance, the enable/disable field allows debugging to be turned on and off, the

single-step field allows a break after each instruction, which allows a user to view the effects of each instruction on the system, and the run field allows a breakpoint to be enabled/disabled for a specific address. So, even though an instruction should break because it corresponds to an address, the user may decide skip that break and allow the processor to continue running for whatever reason. As a result, in order to increase flexibility and functionality, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Glew such that the debug processor manipulates at least three debug register bit fields of at least one processor control status register, the at least three register bit fields comprising a run field, a single step field and a debug enable field.

e) Glew has further not taught that the debug process accesses an internal status of one or more of the plurality of processors by utilizing at least one of a load to instruction RAM (LDTI) instruction and a load from instruction RAM (LDFI) instruction. However, the examiner asserts that it is well known in the art to load values into registers during debugging (using an "LDFI" instruction) so the user can see how the system reacts to certain data. The user can then monitor the reactions and responses by the system to diagnose any potential problems with the code. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Glew such that the debug process accesses an internal status of one or more of the plurality of processors by utilizing a load from instruction RAM (LDFI) instruction (the memory being memory 17 of Fig.1, which is known in the art to hold both instructions and data).

12. Referring to claim 2, Glew in view of IBM and further in view of Deng has taught the apparatus of claim 1, wherein said at least one breakpoint bit field allows a breakpoint to be one of set and not set for each of said one or more instructions. See IBM.

13. Referring to claim 5, Glew in view of IBM and further in view of Deng has taught the apparatus of claim 1, wherein said single step field allows a set of instructions to each be single-stepped through one cycle at a time. See Fig.16, field 206, and column 5, lines 29-33, of Deng.

14. Referring to claim 6, Glew in view of IBM and further in view of Deng has taught the apparatus of claim 1, wherein said debug enable field one of enables and disables a debug mode. See Fig.16, field 208, of Deng.

15. Referring to claim 7, Glew in view of IBM and further in view of Deng has taught an apparatus as described in claim 1. Glew has not explicitly taught that the LDTI instruction loads content of a register in a processor of the plurality of processors into an instruction memory coupled to the processor via a bus. However, the examiner asserts that it is well known that store instructions are used in debugging to obtain register content and store it (or load it) into a memory so that they can be monitored by the user. The monitoring will allow the user to determine whether any bugs exist within the code based on whether the data obtained and stored (or loaded) into the memory is expected data. As a result, in order to assist in debugging, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Glew to include an LDTI instruction, wherein the LDTI instruction loads content of a register in a processor of the plurality of processors into an instruction memory coupled to the processor via a bus, the memory being component 17 in Fig.1 (which is known in the art to contain both instructions and data).

16. Referring to claim 8, Glew in view of IBM and further in view of Deng has taught an apparatus as described in claim 7, wherein the LDFI instruction loads the content of the instruction memory into the register coupled to the processor. See the rejection of claim 1(e).

17. Referring to claim 18, Glew has taught an apparatus comprising a machine-readable medium containing instructions which, when executed by a machine, cause the machine to perform operations comprising:

a) adding at least one breakpoint bit field comprising a single bit directly to each of a plurality of instructions to execute on a plurality of processors. See Fig.3, at least component 200, Fig.4, and column 6, lines 24-38. Also, see Fig.1, Fig.3, and Fig.4. Note that a superscalar processor includes N processing elements to execute up to N instructions in parallel (this is the inherent nature of a superscalar system). Hence, the N processing elements are the plurality of processors, as they each process data.

b) Glew has not taught that the breakpoint bit field enables a user of the apparatus to set a breakpoint based on an address of an instruction of the one or more instructions that the breakpoint bit field is attached without having to perform an address comparison. However, IBM has taught such a concept. See Fig.2 and the disclosure. Essentially, IBM allows a user to attach a breakpoint bit to each instruction without performing address comparison, thereby saving hardware, delay, and power associated with the comparison circuitry. One of ordinary skill in the art would have recognized that the controller of Glew (Fig.3, component 200, and column 6, lines 24-38) would be replaceable by the controller of IBM without "breaking" Glew because both Glew and IBM teach generating single breakpoint bits and attaching each bit to an instruction. In Glew, when the instructions are to be decoded and executed, all that is looked for is an instruction having a breakpoint bit. It doesn't matter whether that extra bit was attached in the manner specified by Glew or in the manner specified by IBM. Therefore, in order to maintain a system capable of debugging a system with multiple processors while reducing

hardware, delay, and power consumption associated with Glew's address comparison circuitry, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Glew such that the controller executes a debug process that attaches at least one breakpoint bit field directly to one or more instructions of the plurality of processors, the breakpoint bit field to enable a user of the apparatus to set a breakpoint based on an address of the particular instruction without having to perform an address comparison.

c) Glew has further not taught manipulating at least three debug register bit fields of at least one processor control status register, the at least three register bit fields comprising a run field, a single step field and a debug enable field, each comprising a single bit. However, Deng has taught such a concept. See Fig.16 of Deng and note field 208, which is a debug enable field, field 206, which is a single-step field, and fields BE1, BE2, BE3, or BE4, any one of which is a run field. These bits allow for increased functionality and flexibility in debugging. For instance, the enable/disable field allows debugging to be turned on and off, the single-step field allows a break after each instruction, which allows a user to view the effects of each instruction on the system, and the run field allows a breakpoint to be enabled/disabled for a specific address. So, even though an instruction should break because it corresponds to an address, the user may decide skip that break and allow the processor to continue running for whatever reason. As a result, in order to increase flexibility and functionality, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Glew such that the debug processor manipulates at least three debug register bit fields of at least one processor control status register, the at least three register bit fields comprising a run field, a single step field and a debug enable field.

d) Glew has further not taught accessing an internal status of one or more of the plurality of processors by utilizing at least one of a load to instruction RAM (LDTI) instruction and a load from instruction RAM (LDFI) instruction. However, the examiner asserts that it is well known in the art to load values into registers during debugging (using an "LDFI" instruction) so the user can see how the system reacts to certain data. The user can then monitor the reactions and responses by the system to diagnose any potential problems with the code. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Glew such that the debug process accesses an internal status of one or more of the plurality of processors by utilizing a load from instruction RAM (LDFI) instruction (the memory being memory 17 of Fig.1, which is known in the art to hold both instructions and data).

18. Referring to claim 19, Glew in view of IBM and further in view of Deng has taught the apparatus of claim 18, further containing instructions which, when executed by a machine, cause the machine to perform operations including: determining a state of said breakpoint bit, and setting a breakpoint for an instruction if it is determined that said state of said at least one breakpoint bit field is set. See IBM and Glew, column 6, lines 24-38.

19. Referring to claim 21, Glew in view of IBM and further in view of Deng has taught the apparatus of claim 19, further containing instructions which, when executed by a machine, cause the machine to perform operations including: determining a state of a run field bit, and running a set of instructions if said state of said run field bit is set, and stopping a set of instructions if said state of said run field bit is not set. See Deng, Fig.16, and note that if any first field of four-bit field 202 is enabled, then a set of instructions is stopped when the address associated with the first field. The instructions stopped are the instruction associated with the address and all

subsequent instructions. If the first field is disabled, then the set of instructions will run because the address associated with the first field does not result in a breakpoint.

20. Referring to claim 22, Glew in view of IBM and further in view of Deng has taught the apparatus of claim 21, further containing instructions which, when executed by a machine, cause the machine to perform operations including: determining a state of a single step bit, single-stepping through a set of instructions for a cycle if said state of said single-step bit is set. See Fig.16, field 206, and column 5, lines 29-33, of Deng.

21. Referring to claim 23, Glew in view of IBM and further in view of Deng has taught an apparatus as described in claim 18.

a) Glew, as modified, has further taught that the LDFI instruction loads content of said instruction memory into the at least one register. See the rejection of claim 18(d).

b) Glew has not taught that the LDTI instruction loads content of at least one register into an instruction memory. However, the examiner asserts that it is well known that store instructions are used in debugging to obtain register content and store it (or load it) into a memory so that they can be monitored by the user. The monitoring will allow the user to determine whether any bugs exist within the code based on whether the data obtained and stored (or loaded) into the memory is expected data. As a result, in order to assist in debugging, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Glew to include an LDTI instruction, wherein the LDTI instruction loads content of a register in a processor of the plurality of processors into an instruction memory coupled to the processor via a bus, the memory being component 17 in Fig.1 (which is known in the art to contain both instructions and data).

22. Referring to claim 24, Glew has taught a method comprising:

a) adding at least one breakpoint bit field comprising a single bit directly to each of a plurality of instructions to execute on a plurality of processors. See Fig.3, at least component 200, Fig.4, and column 6, lines 24-38. Also, see Fig.1, Fig.3, and Fig.4. Note that a superscalar processor includes N processing elements to execute up to N instructions in parallel (this is the inherent nature of a superscalar system). Hence, the N processing elements are the plurality of processors, as they each process data.

b) Glew has not taught that the breakpoint bit field enables a user to set a breakpoint based on an address of an instruction of the one or more instructions that the breakpoint field is attached without having to perform an address comparison. However, IBM has taught such a concept. See Fig.2 and the disclosure. Essentially, IBM allows a user to attach a breakpoint bit to each instruction without performing address comparison, thereby saving hardware, delay, and power associated with the comparison circuitry. One of ordinary skill in the art would have recognized that the controller of Glew (Fig.3, component 200, and column 6, lines 24-38) would be replaceable by the controller of IBM without "breaking" Glew because both Glew and IBM teach generating single breakpoint bits and attaching each bit to an instruction. In Glew, when the instructions are to be decoded and executed, all that is looked for is an instruction having a breakpoint bit. It doesn't matter whether that extra bit was attached in the manner specified by Glew or in the manner specified by IBM. Therefore, in order to maintain a system capable of debugging a system with multiple processors while reducing hardware, delay, and power consumption associated with Glew's address comparison circuitry, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Glew such that the

controller executes a debug process that attaches at least one breakpoint bit field directly to one or more instructions of the plurality of processors, the breakpoint bit field to enable a user of the apparatus to set a breakpoint based on an address of the particular instruction without having to perform an address comparison.

c) Glew has further not taught manipulating at least three debug register bit fields of at least one processor control status register, the at least three register bit fields comprising a run field, a single step field and a debug enable field each comprising a single bit. However, Deng has taught such a concept. See Fig.16 of Deng and note field 208, which is a debug enable field, field 206, which is a single-step field, and fields BE1, BE2, BE3, or BE4, any one of which is a run field. These bits allow for increased functionality and flexibility in debugging. For instance, the enable/disable field allows debugging to be turned on and off, the single-step field allows a break after each instruction, which allows a user to view the effects of each instruction on the system, and the run field allows a breakpoint to be enabled/disabled for a specific address. So, even though an instruction should break because it corresponds to an address, the user may decide skip that break and allow the processor to continue running for whatever reason. As a result, in order to increase flexibility and functionality, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Glew such that the debug processor manipulates at least three debug register bit fields of at least one processor control status register, the at least three register bit fields comprising a run field, a single step field and a debug enable field.

d) Glew has further not taught accessing an internal status of one or more of the plurality of processors by utilizing at least one of a load to instruction RAM (LDTI) instruction and a load

from instruction RAM (LDFI) instruction. However, the examiner asserts that it is well known in the art to load values into registers during debugging (using an "LDFI" instruction) so the user can see how the system reacts to certain data. The user can then monitor the reactions and responses by the system to diagnose any potential problems with the code. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Glew such that the debug process accesses an internal status of one or more of the plurality of processors by utilizing a load from instruction RAM (LDFI) instruction (the memory being memory 17 of Fig.1, which is known in the art to hold both instructions and data).

e) Glew in view of IBM and further in view of Deng have taught that at least one breakpoint bit field is an additional field directly added to each processor instruction. See IBM (the disclosure and Fig.2) and Glew, column 6, lines 24-38.

23. Referring to claim 25, Glew in view of IBM and further in view of Deng has taught the method of claim 24, further comprising determining a state of a breakpoint bit, and setting a breakpoint for an instruction if it is determined that said state of said breakpoint bit is set. See IBM and Glew, column 6, lines 24-38.

24. Referring to claim 26, Glew in view of IBM and further in view of Deng has taught the method of claim 24, further comprising running a debug process on a host device, and entering debug commands through a graphical user interface. See Deng, column 1, line 50, to column 2, line 9. Note that if a user is stepping through a program, then that user must enter a step command.

25. Referring to claim 28, Glew in view of IBM and further in view of Deng has taught the method of claim 24, further comprising: determining a state of a single-step bit, entering

commands for single-stepping through a set of instructions for a cycle if said state of said single-step bit is set. See Fig.16, field 206, and column 5, lines 29-33, of Deng.

26. Referring to claim 29, Glew in view of IBM and further in view of Deng has taught a method as described in claim 24.

a) Glew, as modified, has further taught that the LDFI instruction loads content of said instruction memory into the at least one register. See the rejection of claim 24(d).

b) Glew has not taught that the LDTI instruction loads content of at least one register into an instruction memory. However, the examiner asserts that it is well known that store instructions are used in debugging to obtain register content and store it (or load it) into a memory so that they can be monitored by the user. The monitoring will allow the user to determine whether any bugs exist within the code based on whether the data obtained and stored (or loaded) into the memory is expected data. As a result, in order to assist in debugging, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Glew to include an LDTI instruction, wherein the LDTI instruction loads content of a register in a processor of the plurality of processors into an instruction memory coupled to the processor via a bus, the memory being component 17 in Fig.1 (which is known in the art to contain both instructions and data).

Response to Arguments

27. Applicant's arguments filed on June 2, 2008, have been fully considered but they are not persuasive.

28. Applicant argues the novelty/rejection of claim 1 on page 13 of the remarks, in substance that:

> "The Examiner equates the run field of claim 1 with field 202 (BE1, BE2, BE3, or BE4) of Figure 16 of Deng. However, field 202 is actually a plurality of breakpoint enable fields spanning 4 bits, not a run field spanning a single bit. The run field of claim 1 is not the same as the breakpoint field in claim 1. The disclosure of Deng in Figure 16 provides for multiple breakpoint enable bits for particular instructions, rather than for a run field of a single bit that would cover execution of instructions in an entire processor. As such, Deng does not disclose or suggest the cited feature of claim 1."

29. These arguments are not found persuasive for the following reasons:

a) The examiner asserts that any one of the bits BE1, BE2, BE3, and BE4, qualifies as a run field. Field 202 as a whole does not need to be viewed as the run field. Instead, each subfield within field 202 qualifies as the run field. Hence, Deng has taught debug enable, single-step, and run fields, each comprising a single bit.

Conclusion

30. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to DAVID J. HUISMAN whose telephone number is (571)272-4168. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/David J. Huisman/
Primary Examiner, Art Unit 2183
August 4, 2008